Section 4—Technology Model

The Technology Model contains information related to the hardware, system software, and network components needed to support the CTIS CDMS. The Technology Model comprises the technology infrastructure architecture, key technical factors, system management model, and the development/testing physical architecture.

The technology infrastructure architecture identifies the technologies on which the CTIS CDMS will be deployed and describes their functionality.

The key technical factors present information about the key technical drivers that drive system design, and they identify potential technical risk areas for the CDMS system.

The system management model identifies system strategies for backup and recovery, system tuning, and system administration and maintenance.

The development/testing physical architecture identifies the hardware configurations for development and the tools used in the development process.

4.1 Technology Infrastructure Architecture

Infrastructure refers to the hardware, software, and systems software that serve as the foundation on which the CDMS will be built. This section addresses the infrastructure components of the CDMS system as well as the logical organization of these components.

4.1.1 Three-Tier Architecture

The CDMS will be constructed as a three-tier system. A tier is a vertical partitioning of the system into independent and weakly coupled groupings. A tier is a logical concept that will map to a machine, part of a machine, or a set of machines. The three tiers are presentation, application, and data. The tiers are tied together via a common network and an interprocess communication mechanism that allows the tiers to interoperate efficiently. These tiers should be viewed as architectural concepts, not physical implementation restrictions. The tiers do not constrain the physical partitioning of the system. The three tiers assist in realizing scalability, usability, reliability, and interoperability. Figure 4 illustrates a three-tier system.

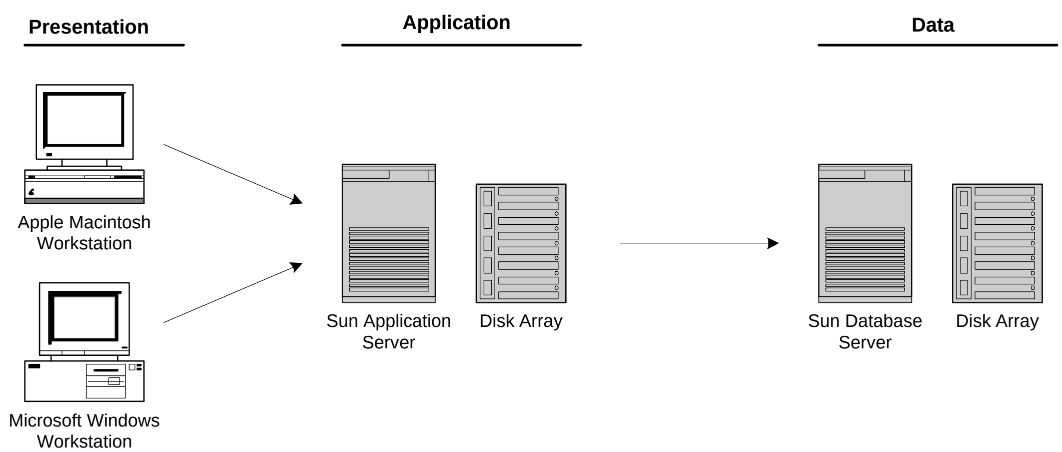

Figure 4. Three-Tier Architecture

4.1.2 Tier Service Layers

Each tier is composed of service layers. Layering is a horizontal partition of the tier. The layers build upon services of lower layers, but a layer can access services of any lower layer.

There are three layers within each tier: the technology-focused and business-independent layers are hardware and base technologies; the business-focused layer is the application component; the application component may be broken down on a tier into reusable application components and custom application components. The layers are described as follows:

- **Hardware layer**—The hardware layer is composed of the computer hardware used in the system. This layer provides the computing base for the system.
- **Base technology layer**—The base technology layer is the lowest software layer of the tier containing system technologies supplied by vendors. This layer contains components such as operating systems, databases, and communication protocols. It also contains any vendor-supplied middleware such as application management server software.
- **Application component layer**—The reusable application component sits on top of the base technologies and provides business or application-specific services to applications. The emphasis for these components is reuse. Custom application components are the highest layer in the tier and contain components that implement a specific application function. Figure 5 illustrates the relationship of service layers within a tier.

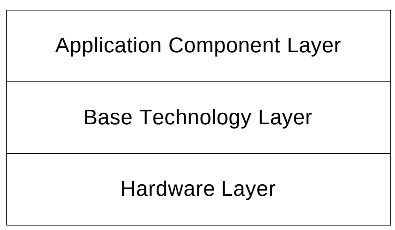

Figure 5. Tier Service Layers

4.2 CTIS Tier/Service Layer Architecture

Figure 6 illustrates the CTIS operation system technical architecture in more detail. The diagram shows both the logical distribution of functionality via tiers and the service layer components comprising each tier.

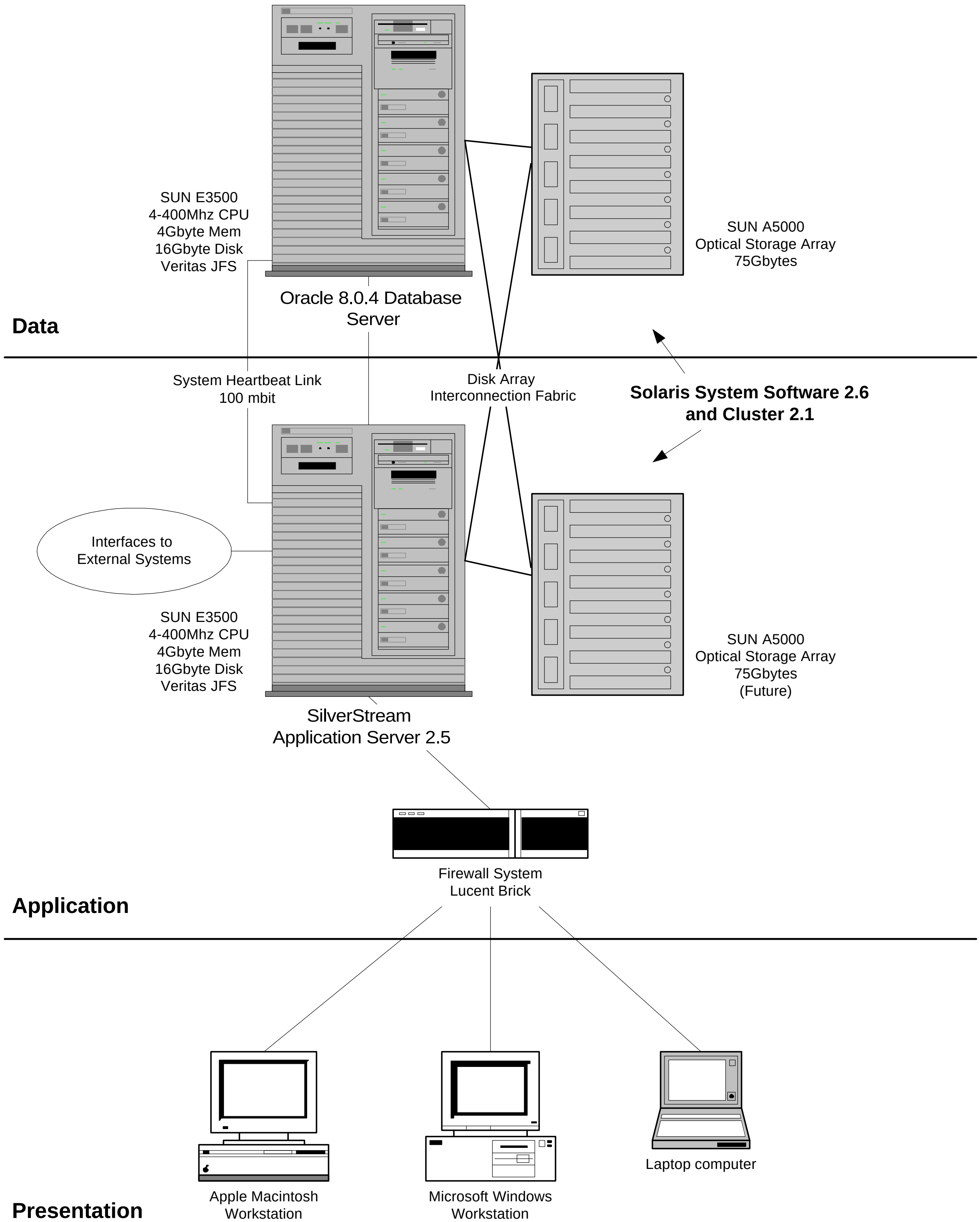

Figure 6. Tier/Service Layer Architecture

The following subsections identify the individual CTIS service components and describe their functionality by tier.

4.2.1 Data Tier Service Components and Functionality

The data level of the infrastructure provides all the services associated with the storage, security, and manipulation of data associated with the CDMS.

Data Tier—Hardware Layer

- **Data server hardware—**Data server hardware for CDMS provides the platform for data storage and data access services. The NCI Information Systems and Technology Branch (ISTB) planned hardware platform will consist of two SUN Enterprise 3500 servers, each with four 400 MHz processors and four gigabytes of memory. The machines will be connected in a cluster configuration to provide the capability for automatic failover of processes running on the other cluster node. The normal operational mode is for one SUN server machine to host the data tier and the other to host the application tier. In the case of machine failure, both tiers can be run on a single server with some service degradation.
- **Data storage hardware—**Data storage hardware provides permanent storage for data. The ISTB plans to provide a high-speed RAID array disk subsystem linked to the cluster via fiber channel. This will provide the highest speed input/output (I/O) capability to enhance disk I/O performance. The amount of data storage will be sufficient to hold extra copies of the database in a backup format so that if the working database becomes corrupt, it can be restored and the system returned to a functional state in a rapid manner. The extra disk space is important in this situation because restoring data from disk is much faster than tape.
- **Data server networking hardware—**Networking provides linkage functionality from the data tier to the application tier. The servers will be interconnected with a 100 Mb Ethernet link to provide communication services between cluster nodes, and they will be connected to the NIH campus backbone network at 100Mb speed for fast communications.

Data Tier—Base Technology Layer

- **Data server operating system—**The operating system provides the basic control program for the database server. The operating system must support moderate volume transactional systems in a highly available environment. The operating system and hardware environment will be able to support the requirements for the relational database management system. The hardware cluster will run SUN Solaris 2.6 initially, which will be upgraded to Solaris 7.0 when all devices are supported for the cluster configuration. Solaris supports the multitasking and multithreading as required in a cluster environment and offers support for load balancing across the cluster for enhanced performance. The cluster software allows the system as a whole to detect a failure of either node and restart processes that were running on the failed node to return the system to full function with little or no downtime for the user.
- **Database management system—**The relational database management system controls all functionality associated with the storage and access of data. The RDBMS must

provide high-performance access, security, scalability, high integrity, and high availability services for CDMS data. Oracle version 8.0.4 will be installed and configured for the Solaris environment. Oracle supports the Solaris platform and can be configured to take advantage of the multiple processors and load balancing across the cluster for optimized database performance. Database backup/recovery procedures are addressed in the System Management Module section of this document.

- **Data server communications protocols**—The communication protocol defines how information is transmitted to the application tier. Both the application and database servers will use the underlying TCP/IP protocol to transmit information. Oracle SQLNet and Java Database Connectivity (JDBC) drivers will sit on top of the TCP/IP protocol. The JDBC driver on the application server will translate the JDBC database calls into native Oracle calls for transmission over SQLNet.
- **Data server management tools**—These tools manage components on the data server. The ISTB will be responsible for overall database administration on the data server and will use the administration components that come with the Oracle database server to perform these functions. Additional utilities will be employed to monitor database performance from both the logical and physical level to ensure optimal performance of the RDBMS.

Data Tier—Application Component Layer

- **Database Data Definition Language (DDL)—**This database-supplied construct allows for definition of data structure within the database. These commands will be used to implement the CDMS physical entity relationship diagram (ERD) described in the Data Model section of this document.
- **Database Manipulation Language (DML)—**This database-supplied construct allows for manipulation and access to the data in the database. DML will implement application-level business rules defined in the Application Model section of this document.

4.2.2 Application Tier Service Components and Functionality

The application level of the infrastructure provides all the services necessary to implement the business logic associated with the CDMS.

Application Tier—Hardware Layer

- **Application server hardware**—Application server hardware for CDMS provides the platform for business application logic. The application tier hardware is identical to the data tier server hardware. Please refer to the Data Tier Server Hardware section for a description.
- **Application server networking hardware**—Networking provides linkage functionality for the application tier to the data tier (refer to the Data Tier—Data Server Networking Hardware description) and the presentation tier (refer to the Presentation Tier—Client Networking Hardware description).

Application Tier—Base Technology Layer

- **Application server operating system**—The operating system provides the basic control program for the application server. The operating system must support moderate volume transactional systems in a highly available environment. The application tier operating system is identical to the data tier operating system. Please refer to the Data Tier section for a description.
- **Application server communications protocols**—The communication protocol defines how information is transmitted to/from the application tier to the presentation tier (refer to the Presentation Tier—Client Communication Protocol description) and data tier (refer to the Data Tier—Data Server Communication Protocol description).
- **Application server transactional application server**—Application server software provides the environment for hosting business-specific application components and provides services to these components. The primary job of the application server system is to manage presentation tier connections, host business logic, and direct traffic for incoming client requests to back-end resources as well as provide services to application components. The services provided include connection management, session management, client authentication, and database connectivity. The application server must provide high-performance access, security, scalability, high integrity, and high-availability services for CDMS data transactions.

  The application server for the CDMS is the SilverStream application server. This application server provides all the capabilities listed above plus a complete, visually based integrated development environment for high-developer productivity. Appendix A contains a product comparison matrix for several application servers evaluated by CSC.
- **Application server printing**—Server-side printing enables the printing of large, batch-oriented reports via routing over a network to a print center. This will require an application software print component.
- **Application server management tools**—These tools manage components on the application server. The SilverStream application server is managed by a browser-based application called the Server Management Console (SMC). This application provides administrative access to most application server functions. Using the SMC, the administrator is able to configure database and e-mail server interfaces, perform software license management, monitor server performance statistics, modify server performance characteristics, configure load balancing processing distribution, and review the application server's system log. In addition, the SMC provides access to the application server's security configuration, including server, application, and database security options. The application server's secure socket layer (SSL) configuration is managed through the SMC as well, and it provides access for administering server certificates, certificate authorities, and the encryption algorithm.

Application Tier—Application Component Layer

- **Application components**—These components are hosted within the SilverStream application server and contain most of the CDMS functionality as well as interfaces to the application server software services. The application components are described in the Application Model section of this document.

4.2.3 Presentation Tier Service Components and Functionality

The presentation tier is the channel for users to access all CDMS clinical and administrative functions via presentation logic on user workstations. A requirement for the CDMS was to provide universal client access; therefore, presentation logic is provided via a Web browser interface.

4.2.3.1 Presentation Tier—Hardware Layer

- **Client workstation hardware**—The client hardware provides the platform for the user workstation. The client hardware layer for CDMS will be based on two platforms: Intel-based Pentium and Celeron processor systems and Apple Macintosh PowerPC systems.
- **Client networking hardware**—The networking equipment connects the users to the CDMS application logic on the application tier. CDMS client systems are connected to the DCS local area network (LAN) via 100baseT Ethernet and through 10baseT Ethernet in a minority of locations where 100baseT Ethernet support is not available. Nominal configuration attaches no more than 12 nodes on a network hub, which in turn connects to a 100BaseT Ethernet switch providing access to the NIH gigabit network backbone.
- **Client printing**—Printing provides the means to generate hardcopy output. Clients will have the capability to print to printers attached to workstations via the workstation print capability.

Presentation Tier—Base Technology Layer

- **Client operating system**—The client operating system provides basic control and support functions for system and application software executing on user workstations. Several operating systems are supported on CDMS client systems. For the Apple Macintosh platform, MacOS version 8.1 or higher operating system will be supported. Intel platforms will support Microsoft Windows 95 and Windows 98 operating systems.
- **Client Web browsers**—Client Web browsers will host CDMS presentation software. The preferred browser to support the CDMS application software on the Apple Macintosh platform will be the Netscape Communicator version 4.5 browser. On the Intel platform, the Netscape Communicator version 4.5 browser will be the preferred host. Support will be provided for the Microsoft Internet Explorer version 4.0 browser on Intel platforms and Microsoft Internet Explorer version 4.5 on the Apple Macintosh platform.
- **Client communications protocols—**The communication protocol defines how information is transmitted from the client Web browser to the application tier. All CDMS client systems will communicate using TCP/IP as the basic transmission protocol. Information requests and responses between the client browser and the application server will be supported using the HTTP and HTTPS protocols. The HTTPS protocol, which is an SSL-encrypted data stream protocol, will be used for all data exchange between client and server systems. The HTTPS protocol utilizes the RSA encryption algorithm for encrypting all data communication traversing the HTTPS link.

Presentation Tier—Application Component Layer

- **Application components**—These components are hosted within the Web browser and present the CDMS graphical user interface (GUI) interface for screens and small reports. Presentation layer components will contain some business logic related to data validation and data presentation. The application components are described in the Application Model section of this document.

4.3 Key Technical Factors

This section presents a compilation of key technical factors that were the basis for the system design. Each factor is described and, where applicable, any remaining risk areas are identified.

4.3.1 Infrastructure Security

Protecting CDMS clinical data from unauthorized access was the principle driver for the design of the infrastructure architecture. The system has to be protected from external threats (e.g., hackers) who try to access clinical data or damage the system as well as unauthorized internal use. In addition, good security architecture must enable users to access the legitimate data and functionality they need. The major constraint within the DCS environment is that the DCS infrastructure is not currently protected by a firewall system that creates a “closed” and therefore trusted communication system. Given that constraint, the CDMS security goals were as follows:

- Authentication to verify the identity of a system user
- Access control that defines which users, or groups of users, have access to which resources (such as programs and data)
- Information transmission between the client tier and the application server tier in a secure manner
- Data integrity to ensure that users cannot damage data and to safeguard data assets
- Security for the application server (application tier) and database server (database tier) behind a firewall in a trusted and secure environment

These security goals were resolved with the following system design features:

- Support an encrypted link between the workstation (client browser) and application server. The primary mechanism for securing communications between the client and the server is to use SSL services. This means that all communication between the client workstation and the application server is encrypted. The communication link between the application server and database server is behind the firewall and is therefore secure. This configuration provides the necessary security for keeping clinical data safe and secure and for allowing the communication between the application server and database server to be optimized for performance.
- The application server allows for creating secure interfaces to other systems at the NIH/NCI such as the Medical Information System (MIS), CHCS, and the Clinical Data Repository. It will be necessary to analyze these interfaces and create the specifications for securing the data transfer connections.

- This configuration, however, does not currently address network printing. More investigation into this issue will be required, specifically focusing on the capability of the application server to support secure printing.

4.3.2 Availability

The availability of the system for the users during the primary period from 7:00 a.m. to 7:00 p.m. is critical. Migrating from single user systems to a centralized system increases the need for a reliable and available system. CSC has chosen to recommend a hardware design that addresses these issues in a number of ways.

The chosen architecture is a two-node, dual processor (two CPUs per node) SUN Solaris Server System in a fault-tolerant configuration. Fault-tolerant configuration means that there is a heartbeat link between the two nodes to detect node failure. In the case of node failure, the remaining node can recover the services of a failed node. The two nodes together form what is known as a cluster. This architecture provides for:

- A flexible cluster hardware design (multiple processing nodes) to maintain overall system performance and response time for the various types of database processing
- Maximum availability and reliability in a processing environment
- Automatic failover for a cluster node that becomes inoperable
- Software scripts to restart any software services from the downed processing node
- The ability to run software on multiple cluster nodes to perform load balancing and optimization
- A cluster configuration that will support an application server that is a multithreaded process handling multiple requests over multiple machines, thus giving application servers very good availability characteristics

4.3.3 Scalability

A clustered system architecture and use of system tiers is inherently scalable by the basic nature of the configuration, as follows:

- Each node of the cluster is a multiprocessor machine. Software processes can run on multiple processors to scale up to increased processing requirements and optimize load balancing and improved throughput for performance. The node can be vertically scaled by the ability to add CPUs and memory to a single node.
- The cluster can be horizontally scaled by adding nodes. This expandable hardware system architecture capability can meet increased processing needs as the number of studies, users, and amount of data increase over time or new applications are added to the system.
- The application tier contains the application server, which is designed to be very scalable in performance for the number of users supported. They are designed to handle high-speed, high-throughput, and high-volume transactions and are normally optimized to support users or transactions numbering in the thousands. Therefore, supporting the NCI

environment—which numbers in the hundreds of users—should not be an extreme burden for the application server.

4.3.4 Performance

System response time is very important to the users because it forms their perception of the overall system. The clustered system architecture will allow CSC and NCI to optimize and tune the system to provide the maximum performance with good response time for users. Application design must take into account system performance as a design driver.

CSC plans to work with the NCI ISTB to employ a series of performance analysis tools (described in more detail in the System Management section of this document) to provide a proactive monitoring of the system to uncover any performance bottlenecks before they become apparent to the user community.

CSC has discussed the sizing of the computing facility provided by the NCI ISTB for the DCS CTIS system. In general, there is an agreement to use the strategy of oversizing the system to ensure good performance and to maintain that strategy throughout the development and deployment of the system.

The SSL involves some significant overhead in performing Web transaction encryption; however, increasing memory and CPU at the server level can mitigate this performance impact.

4.3.5 Interface to External Systems

Post Release 1, CTIS will be required to interface with a number of NIH external systems. Not much system analysis or design work has been performed for external interfaces to date. However, in general, application servers facilitate the creation of interfaces to external data sources that are required for CTIS. More detail will be required to identify the support requirements of the external interfaces before their design.

4.3.6 Multiplatform Requirements

Beside security, the other major driver for the system is the requirement to run in a multiplatform environment. CTIS must support the Macintosh platform as well as the Windows PC platform. Support for the Macintosh was critical because of the number of users already using this platform. The process to switch to the PC would involve far more than only the cost of changing the platform. User training, scheduling, disruption of current functions, and general acceptance would have presented considerable obstacles.

The best way for the CTIS architecture to address this issue is to create the application using a Web-based interface, which is a universal client interface. Design technologies involved in this type of client presentation include Web browsers, HyperText Markup Language (HTML), Java, and JavaScript. Although these technologies are known as universal, care must be taken during system design, implementation, and testing because clients are not completely universal. Differences exist between platforms, browsers, and versions within browsers. CSC is sensitive to these slight variations and will test applications in all user environments before system release. For example, the rendering of GUI elements such as a font can vary slightly. Elimination of hard-coded text sizes and position mitigates this problem (i.e., letting the elements scale to their natural size within the confines of the running host platform). Testing on the platform verifies appropriate rendering. Some fonts will render slightly differently and others more differently; limiting through testing to applicable fonts for deployment is possible.

There is some technical concern with the Macintosh support for Java in the browser or in standalone mode. Current industry literature suggests that the Java runtime may be unstable in certain circumstances. CSC's investigation of the problem revealed the following information: the Macintosh platform run-time execution of Java code is not quite as stable as the PC run-time execution of Java code because of the Macintosh Java Virtual Machine (JVM). A JVM is the environment (software environment) that the Java code runs within. The Apple version of the JVM is called the Macintosh Runtime Java (MRJ). This "instability" is evidenced when complex Java classes are run under the JVM, either in a browser JVM or a JVM running over the operating system, and the class behaves unpredictably (e.g., a tab control, when tabbed, does not function properly).

The MRJ is "one-step" behind in terms of support for the Java Development Kit (JDK). The JDK is basically a set of software "pieces" that extend the core Java platform software provided by the SUN Corporation and makes the Java language a useful programming language. SUN controls the JDK and periodically releases new versions of JDK with more useful and powerful tools. As SUN releases each version of JDK, vendors supplying Java software products step up to meet the new JDK. The latest SUN release of JDK—now known as the Java Platform—is called the Java 2 Platform. It used to be known as JDK 1.2.

Java 2 Platform was released in late 1998. Few, if any, vendors have stepped up to that release. The MRJ currently implements JDK 1.1.6. Most vendors are at 1.1.7. CSC e-mailed communication to Peter Lowe, Director, Mac OS, Worldwide Product Marketing and received the following response. "Apple will ensure that Java support on the Mac OS moves forward in sync with developer and customer adoption. Our primary focus today is on continuing to improve the Java 1.1 Platform as we investigate and monitor Java 2." This statement sounds like continuing support, but currently no schedules are available from Apple.

4.3.7 Manageability

The CTIS architecture needs to be a cost-effective for DCS. To this end, the hardware and systems software provide a high degree of manageability through systems management tools that provide for centralized administration and, in the future, that are remotely accessible through a browser. The SUN Solaris 2.6 or 7 (in the future) operating system administrative tool is the Admin Tool; the Oracle RDBMS tool for database administration is the Enterprise Manager; and, the application server administration tool is the SMC.

The CDMS application itself will be highly extensible through the data dictionary definition of clinical terms. From this application core, the CDMS will generate database schema necessary to store clinical data for studies, thereby eliminating the need for application clinical database schema maintenance via a database administrator. The user interface to the data dictionary will be highly interactive and enable real-time access to updating the data dictionary.

4.4 System Management Model

The reliability and availability of the CDMS production system depends on strategies and support processes defined to manage the DCS production system. The following are a list of systems management and support processes that are relevant to the CDMS system and that should be planned for to support the operational system.

These suggestions for system administration, backup, and recovery of database and system tuning are being made as if CSC was responsible for the systems so as to identify a complete set of procedures for these functions. The NCI ISTB is essentially providing a computing service to DCS. NCI and CSC have worked together to create a fault-tolerant and scalable computer system architecture. However, the ISTB will be responsible for system backup, system performance tuning, and overall system administration.

The plan for system and database backups includes the capability to store backup data on an automated, large capacity, tape drive jukebox. The device will hold multiple tape libraries and have a number of tape drives. Again, the system administration staff at the ISTB will actually be performing the backups on a regular schedule.

4.4.1 Backup and Recovery

System data must be protected from accidental loss or deliberate destruction. This requires the capability to make backup copies of the data periodically and to restore files from the backup data if the primary file is lost or corrupted. Since all CDMS data will be stored in the Oracle RDBMS, the extensive capabilities of Oracle backup and recovery need to be put into place.

There are two purposes in creating the backups for the system and the database. One is to be able to recover from a system failure, which requires the system to be reloaded and rebuilt, and the other primary goal is to recover the database if for some reason the current version becomes corrupt.

One method is to back the system and database files to tape, and the other is to have an export of the database available on extra disks within the system. The main goal for this configuration is to maximize the speed of restoring the entire database, if necessary. Essentially restoring from disk will be an order of magnitude faster than restoring from tape.

One of CDMS' design goals is to be able to backup and recover data quickly, with maximum data recoverability and minimal impact to users. CDMS will achieve this goal using two mechanisms.

First, the database files will be located on a high-speed RAID disk array. The RAID array will mirror the data on multiple disks. Therefore, the RAID has some inherent recoverability in that if one of the disks becomes damaged, it can automatically be replaced with the hot spare standby disk, and its data can be restored using the remaining disks. However, if more than one disk in the array becomes damaged, then RAID technology cannot recover the lost data.

Second, CDMS will use the Oracle server's archiving function. When the database is first created, CDMS will make a full offline database backup. A copy of this backup will be kept on disk for fast recoverability and on tape, and kept offsite, in case of complete loss of the server computer and/or facility (e.g., such as a fire). CDMS will then put the database in archive log mode. This causes the Oracle Server to keep a record of all committed database transactions in archived redo logs. In the event of a media failure (e.g., a disk crash), these archived redo logs can be used in conjunction with the database backup files to recover the committed data up to the time of the failure.

To minimize recovery time, it is desirable to minimize the number of archived redo logs that must be restored. CDMS will accomplish this by establishing a new, full database backup on a periodic basis. The system is not available to users during a full database backup; therefore, if the system can be taken down periodically for scheduled non-prime time hours maintenance and backup, then CDMS will perform a full offline backup at those times. There should be sufficient time during the evening hours or on weekends to create a full database backup to minimize the need to use the Oracle server's online, incremental backup capability to backup each tablespace in the database one at a time, which is much less efficient.

Note that recovery after an abnormal shutdown (e.g., power failure) is automatically handled by the Oracle server as part of the normal database startup procedure. All transactions that were committed before the shutdown will be automatically recovered.

4.4.2 Data Dictionary Maintenance

A key concept for DCS is the use of standardized data to describe and codify clinical events. This standardized data is maintained in a common data dictionary in the CDMS. The dictionary information supports all the DCS business processes surrounding standardized data. An application is available for maintenance of this dictionary. This maintenance application will enable users with certain security levels to maintain standard data. The standardized data should always be maintained through use of the application, never through online database editing.

4.4.3 System Tuning

System tuning for CDMS centers on two major areas:

- Database tuning
- Solaris platform system performance monitoring

Database Tuning

The performance of the Oracle database should be monitored regularly. There are a number of factors and techniques that affect the performance of the database and, therefore, the performance of CDMS. In general order of importance, these include physical database file placement, application tuning, memory usage, and I/O.

With regard to physical database file placement, some files in an Oracle database experience a high frequency of inserts, updates, and deletes, and other files are typically more stable. Files supporting volatile tablespaces, such as rollback segments, sort areas, and redo logs, should be kept on physical disk drives separate from more stable tablespaces, such as the data dictionary. Files supporting indexes should be kept separate from the files supporting the corresponding application data. This segregation of files will minimize disk contention and thereby maximize system performance. Note that the CDMS application may dynamically create new tables as new protocols and data standards evolve. Concurrently, physical restructuring or redistribution of data may be needed to improve performance.

With regards to application tuning, Oracle can use indexes to speed the process of locating and retrieving specific data entries. CDMS will automatically create indexes on the primary and foreign keys in the clinical data tables; these will typically include fields such as patient ID, study ID, and event date. If users frequently search and/or sort the data on other fields, then it would be advantageous to create indexes on those fields to improve performance.

With regards to memory usage, Oracle retains data in memory so as to reduce disk I/O. This data includes SQL and PL/SQL statements used by the application, database-level data dictionary information, and application data. Usage of the Oracle shared pool and database buffer cache should be monitored to ensure that they are appropriately sized. If these objects are too small or too large, then system performance can be adversely affected.

There are several factors that affect how efficiently Oracle performs I/O. These include size, number, and placement of rollback segments and redo logs, migration of updated/expanded data records, and efficient use of data blocks. These items can be monitored and tuned to maximize system performance.

Solaris Platform System Performance Monitoring

This activity should measure system performance factors (response time) and utilization factors to provide input for system tuning and capacity planning. From a utilization standpoint, the monitoring of the hardware platform running should include:

- CPU
- Memory
- Disk storage (including I/O throughput, amount of free space, and utilization of I/O per disk)

The additional information per disk is an important consideration for database tuning. The general concept is to spread I/O work for the system across as many disks as possible and to eliminate a potential bottleneck of having one disk with the highest utilization and the other disks relatively inactive.

Monitoring CPU and memory is important from the standpoint that we want to ensure that the system has adequate performance and good response time for the number of users, Web server, application server, and database server processes running on the machine. Monitoring CPU and memory will allow the system administrator to detect if all CPUs are busy or the physical memory has been exhausted, thereby forcing the system to perform excessive paging that will ultimately slow down user response time.

4.4.4 System Administration

System administration for CDMS primarily will involve administering the SUN Solaris platform on which the data and application tiers are hosted. The Information Systems Technology Branch (ISTB) will apply its standard methods for performing system administration and performance monitoring to ensure proper operation of the system and optimal performance. The ISTB has personnel familiar with the SUN environment because it is the ISTB’s standard platform and will employ the administration capabilities inherent in Solaris to monitor and administer the system. Currently, the ISTB is running Solaris version 2.6 and will migrate to Solaris 7 when it becomes stable to take advantage of the browser administration interface.

The clustered server system is running several processes and applications that have been described above:

- Oracle database
- Netscape Web server
- SilverStream application server

Users will not have accounts on the system nor will they have direct access to system services such as file space, etc. Therefore, system administration is limited to the installation, configuration, and monitoring of the services and processes described above.

4.5 Development/Testing Physical Architecture

This section describes the development and test environment for CDMS.

The development facility should be running identical hardware and software versions for the operating system, Oracle database, and SilverStream. Our goal is to have the environments identical to minimize any issues related to migrating from the development to the production environment. The development hardware will be housed at the CSC development site at the Shady Grove facility in Rockville, MD.

NCI ISTB discussed the planned hardware for the CSC development and decided it will consist of 1 SUN Enterprise 450 server, with one 400MHz processor and 1 gigabyte of memory. The machine will run SUN Solaris 2.6 initially and will be upgraded to Solaris 7.0 when the ISTB migrates to that operating system or plans to have the production facility running that level of the operating system.

CSC plans to have a high-speed RAID array disk subsystem linked to the server via a small computer system interface (SCSI) channel. This will provide the high-speed I/O capability to enhance disk I/O performance. The amount of data storage will be sufficient to hold extra copies of the database in a backup format so that, if the working database becomes corrupt, it can be restored and the system returned to a functional state in a rapid manner. The extra disk space is important in this situation because restoring from disk is much faster than tape.

In addition to the hardware and software mentioned above, other tools will be used to simplify various development and testing processes:

- **Data modeling—**Oracle Designer/2000 is used to generate the logical data model for the data dictionary and related schema and can be used to generate the data tier DDL for it.
- **Configuration/problem tracking tool**—CSC plans to use Intersolv’s PVCS for configuration management. This tool allows users to check modules in and out of the configuration database for modifications and saves the previous versions for tracking control. It allows modules to be combined into releases to form libraries and distribution packages so that the system can be tracked as an integrated unit. PVCS supports the concept of multiple releases and version control for all elements under its control.
- **Application server integrated development environment—**The SilverStream application server provides a development environment that enables software developers to develop application components. The SilverStream development environment consists of a development server and the SilverStream designer client. The designer client allows developers to build Web application components. The designer client is a visual

development environment that enables the development of dynamic Web pages in HTML, JavaScript, Java, and business objects in Java.

- **Automated testing tools—**The Rational Software SQA Suite provides an integrated testing suite of tools for testing Windows and browser-based applications. The SQA Suite is comprised of a manager tool, object testing tool, site check tool, and load test tool. The SQA Manager provides an interface to plan, manage, and analyze all aspects of the testing process. The object-testing tool, called SQA Robot, provides an object-oriented approach to designing tests, generating tests through its recording function, and playback of test scripts. Initially, it is anticipated that the use of test tools will focus on the SQA Manager and SQA Robot, not on the Site Check and Load Test tools, which provide Web site verification and stress testing respectively.
- **Help system development tool—**The help system will be constructed with a tool called RoboHelp from BlueSky software. Help information is constructed in Microsoft Word and then compiled into a number of formats that are cross-platform compatible. Web-style help will be the format of help for the CDMS. The user interface model, described in the Online Help Design section, provides a more detailed description of RoboHelp.
- **User documentation tool—**User documentation will be authored with the Microsoft Word word-processing package. Microsoft Word supports collaborative document construction, screen capture and graphic import, and a wide range of document formatting options. Its tight integration with the RoboHelp help-authoring tool provides for easy deployment of user documentation to an online format.